SUPPLEMENTARY MATERIALS

A Deferred Proofs **15**

A.1 Decomposition of the NTK over layers 15

A.2 Derivation of the Diffusion Equation 16

A.3 Local Cauchy Approximation of the C_{NTK} 17

A.4 Derivation of the Correlation Lengthscale from the Cauchy Approximation 19

A.5 Minimum Value of C_{NTK} 20

A.6 Relating Loss Gradient Variance to the NTK 20

B Experimental Details **21**

B.1 Model Training 21

B.2 Order Parameter Estimation 22

C Occurrence Rates of Phase Transitions **23**

C.1 Impact of Image Features 23

C.2 Impact of Hyperparameters 24

D Comparison with ReLU Activations **26**

E Implications of Local Image Structure on Feature Learning **26**

F Evaluating Fidelity of Approximation **28**

F.1 Local Structure of the NTK 28

F.2 Cauchy Approximation 29

G Additional Experimental Results **29**

G.1 Order Parameter Trajectories for Single Runs 29

G.2 Influence of Hyperparameters on Order Parameter Trajectories 29

A DEFERRED PROOFS

A.1 DECOMPOSITION OF THE NTK OVER LAYERS

Consider a feedforward neural network, denoted by $f(x) = h_L \circ \dots h_1(x)$. We furthermore define:

$$z_l = \theta_l^\top h_{l-1} \tag{30}$$
$$h_l = \sigma(z_l) \tag{31}$$

In this way, we may calculate the parametric gradients as follows:

$$\nabla_{\theta_l} f = \frac{\partial f}{\partial z_l} h_{l-1}^\top \tag{32}$$

$$\text{vec}(\nabla_{\theta_l} f) = \left(I \otimes \frac{\partial f}{\partial z_l}\right) h_{l-1} \tag{33}$$

$$\begin{aligned}
\text{vec}(\nabla_{\theta_l} f(x_i))^\top \text{vec}(\nabla_{\theta_l} f(x_j)) &= h_{l-1}(x_i)^\top \left(I \otimes \frac{\partial f(x_i)}{\partial z_l}^\top\right)\left(I \otimes \frac{\partial f(x_j)}{\partial z_l}\right) h_{l-1}(x_j) \quad (34)\\
&= h_{l-1}(x_i)^\top \left(I \otimes \frac{\partial f(x_i)}{\partial z_l}^\top \frac{\partial f(x_j)}{\partial z_l}\right) h_{l-1}(x_j) \quad (35)\\
&= \left(\frac{\partial f(x_i)}{\partial z_l}^\top \frac{\partial f(x_j)}{\partial z_l}\right)\left(h_{l-1}(x_i)^\top h_{l-1}(x_j)\right) \quad (36)
\end{aligned}$$

The first term in this product defines functional similarity between points, while the second defines representational similarity. Thinking of each term as a separate kernel, the overall layer kernel - ie the product is defined via an AND operation. A similar formula holds for the other layers. The full NTK is then given simply by:

$$\begin{aligned}
K(x_i, x_j; \theta) &= \sum_{l=1}^{L} \left(\frac{\partial f(x_i)}{\partial z_l}^\top \frac{\partial f(x_j)}{\partial z_l}\right)\left(h_{l-1}(x_i)^\top h_{l-1}(x_j)\right) \quad (37)\\
&= \sum_{l=1}^{L} K_l(x_i, x_j) \quad (38)
\end{aligned}$$

In particular, we have:

$$K(x, x; \theta) = \sum_{l=1}^{L} \left|\left|\frac{\partial f(x)}{\partial z_l}\right|\right|_2^2 \left|\left|h_{l-1}(x)\right|\right|_2^2 \tag{39}$$

Following the same logic, the full NTK is defined as an OR over all the layers. For INRs, these layers tend to be frequency separated, so that lower layers correspond to lower frequencies.

A.2 Derivation of the Diffusion Equation

In this section, we study kernels of the form:

$$\begin{aligned}
K(x, x+u) &= A(x) e^{-u^2/2\xi^2(x)} \quad (40)\\
&= 2\pi\xi^2(x) A(x) \mathcal{N}(u; 0, \xi^2(x) I) \quad (41)
\end{aligned}$$

Here, $\mathcal{N}(u; \mu, \Sigma)$ denotes the d-dimensional; multivariate Gaussian Distribution:

$$\mathcal{N}(u; \mu(x), \Sigma(x)) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma(x)}} \exp\left(-\frac{1}{2}(u - \mu(x))^\top \Sigma^{-1}(x)(u - \mu(x))\right) \tag{42}$$

For our case, $d = 2$, and $\Sigma(x) = \xi^2(x) I$. The determinant of the covariance is as follows:

$$\det \Sigma(x) = (\xi^2)^2 \det I = \xi^4(x) \tag{43}$$

We now consider the integral of the following quadratic form:

$$\begin{aligned}
\int du\ (u^\top H u)\, e^{-u^2/2\xi^2(x)} &= 2\pi\xi^2(x) \int du\ (u^\top H u)\, \mathcal{N}(u; 0, \xi^2(x) I) \quad (44)\\
&= 2\pi\xi^2(x) \mathbb{E}_{\mathcal{N}(u;0,\xi^2 I)}[u^\top H u] \quad (45)\\
&= 2\pi\xi^2(x) \text{tr}(H\Sigma(x)) \quad (46)\\
&= 2\pi\xi^4(x) \text{tr}(H) \quad (47)
\end{aligned}$$

Now, let's look at the following Taylor expansion:

$$r(x+u) \approx r(x) + u^\top \nabla_x r + \frac{1}{2} u^\top (\nabla_x^2 r) u \tag{48}$$

When integrating the above in equation 3, the second term vanishes, because it involves a product of symmetric and anti-symmetric functions. Thus, we have

$$\int du\; r(x+u)K(x,x+u) = A(x)\int du\; r(x+u)e^{-u^2/2\xi^2(x)} \tag{49}$$

$$= A(x)\int du\; \left[r(x)e^{-u^2/2\xi^2(x)} + \frac{1}{2}u^\top(\nabla_x^2 r)u\, e^{-u^2/2\xi^2(x)}\right] \tag{50}$$

Leveraging our result for the quadratic term, we have, finally:

$$\int du\; r(x+u)K(x,x+u) = 2\pi\xi^2(x)A(x)r(x) + \pi\xi^4(x)A(x)\text{tr}(\nabla_x^2 r) \tag{51}$$

$$= 2\pi\xi^2(x)A(x)r(x) + \pi\xi^4(x)A(x)\Delta^2 r \tag{52}$$

Thus, the diffusion equation becomes:

$$\dot{r} = -2\pi\xi^2(x)A(x)r(x) - \pi\xi^4(x)A(x)\Delta^2 r$$

A.3 Local Cauchy Approximation of the C_{NTK}

A.3.1 Notation and Derivation

We consider an arbitrary vector valued function $f(x)$, and consider the cosine of the angle between $f(x)$ and $f(x+u)$ for small displacements u. To ease notation, let us make use of the following shorthands:

$$a = f(x) \tag{53}$$

$$b = f(x+u) \tag{54}$$

$$c = b - a \tag{55}$$

$$J = \nabla_x a \tag{56}$$

$$D = \nabla_x ||a||^2 \tag{57}$$

To first order in u, we have:

$$b \approx a + u^\top J \tag{58}$$

$$c \approx u^\top J \tag{59}$$

$$||b||^2 \approx ||a + u^\top J||^2 \tag{60}$$

$$= ||a||^2 + u^\top D + ||u^\top J||^2 \tag{61}$$

Our goal is to discern the local behaviour of the cosine of the angle θ between a and b (as illustrated in Figure 7). To that end, our starting point is the law of cosines:

$$\cos\phi = \frac{||a||^2 + ||b||^2 - ||c||^2}{2||a||\,||b||} \tag{62}$$

$$\approx \frac{2||a||^2 + u^\top D}{2||a||^2}\left(1 + \frac{u^\top D}{||a||^2} + \frac{||u^\top J||^2}{||a||^2}\right)^{-\frac{1}{2}} \tag{63}$$

To proceed, note that, for small scalar ϵ, we have the following identity:

$$(1+\epsilon)^{\frac{1}{2}} \approx 1 + \frac{\epsilon}{2} - \frac{\epsilon^2}{8} \tag{64}$$

Thus:

$$\cos\phi \approx \frac{2||a||^2 + u^\top D}{2||a||^2 + u^\top D + ||u^\top J||^2 - \frac{1}{16||a||^2}(u^\top D)^2} \tag{65}$$

$$\tag{66}$$

For the NTK, where we will have $a = \nabla_\theta f$, $||a||$ is so large that we may ignore the term of order $||a||^{-2}$. We illustrate our approximation in Figure 8.

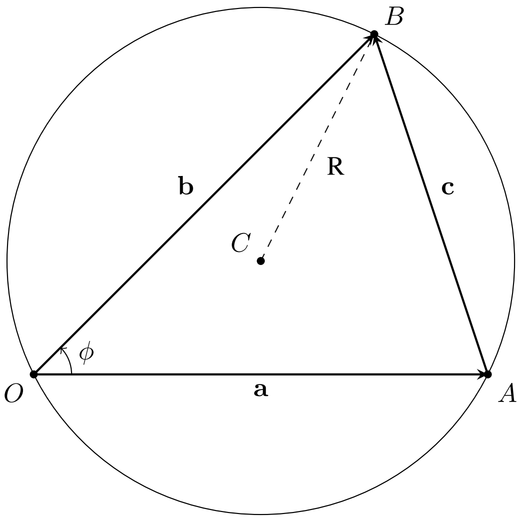


Figure 7: Triangle with vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{b} - \mathbf{a}$, inscribed in a circumcircle.

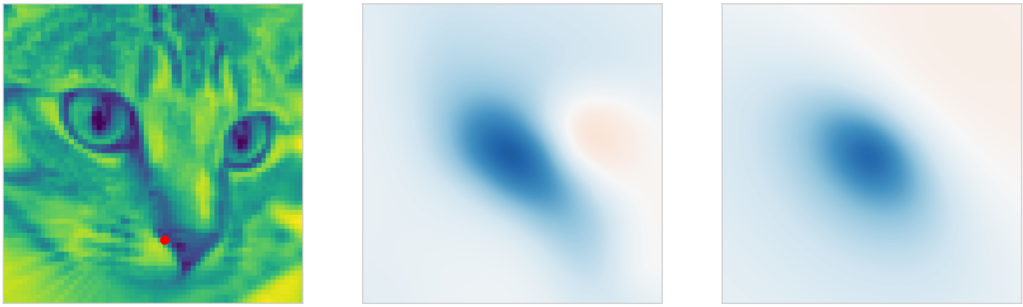


Figure 8: **Cauchy Approximation of the Cosine NTK**. Left: Sample image, and test point $x = A$. Middle: visualization of $C_{NTK}(x, x+u)$ in the vicinity of the point A for small separations u. Right: the Cauchy approximation, capturing both the range, orientation, and the local minima of the true C_{NTK}.

A.3.2 Specialization for Feed Forward Neural Networks

We want to consider the case where, per our previous derivation, $a = \nabla_\theta f$. This procedure is straightforward for the biases. For the weights $W_{ij}^{(l)}$, we have:

$$\frac{\partial f(x;\theta)}{\partial W_{ij}^{(l)}} = \frac{\partial f(x;\theta)}{\partial z_i^{(l)}} h_j^{(l-1)}(x;\theta) \tag{67}$$

Therefore:

$$\frac{\partial f^2(x;\theta)}{\partial x_m \partial W_{ij}^{(l)}} = \frac{\partial^2 f(x;\theta)}{\partial x_m \partial z_i^{(l)}} h_j^{(l-1)}(x;\theta) + \frac{\partial f(x;\theta)}{\partial z_i^{(l)}} \frac{\partial h_j^{(l-1)}(x;\theta)}{\partial x_m} \tag{68}$$

$$\triangleq (J_z^{(l)})_{im} h_j^{(l-1)} + (J_h^{(l-1)})_{jm} \partial_{z_i^{(l)}} f \tag{69}$$

Before proceeding, let us note that the following holds:

$$\sum_i (J_z^{(l)})_{im}(\partial_{z_i^{(l)}} f) = \frac{1}{2}\partial_{x_m}||\nabla_{z^{(l)}} f||^2 \tag{70}$$

$$\sum_i (J_h^{(l)})_{im} h_i^{(l)} = \frac{1}{2}\partial_{x_m}||h^{(l)}||^2 \tag{71}$$

The covariance matrix in our Gaussian approximation is thus given by:

$$H_W^{(l)} = \sum_{i,j} \frac{\partial f^2}{\partial x_m \partial W_{ij}^{(l)}} \frac{\partial f^2}{\partial x_n \partial W_{ij}^{(l)}} \tag{72}$$

$$= \sum_{i,j} (h_j^{(l-1)})^2 (J_z^{(l)})_{im} (J_z^{(l)})_{in} + (\partial_{z_i^{(l)}} f)^2 (J_h^{(l-1)})_{jm} (J_h^{(l-1)})_{jn} \tag{73}$$

$$+ (J_z^{(l)})_{im} (\partial_{z_i^{(l)}} f)(J_h^{(l-1)})_{jn} h_j^{(l-1)} + (J_z^{(l)})_{in} (\partial_{z_i^{(l)}} f)(J_h^{(l-1)})_{jm} h_j^{(l-1)}$$

$$= ||h^{(l-1)}||^2 J_z^{(l)} J_z^{(l)\top} + ||\nabla_{z^{(l)}} f||^2 J_h^{(l-1)} (J_h^{(l-1)})^\top \tag{74}$$

$$+ \frac{1}{4}\nabla_x ||h^{(l-1)}||^2 \otimes \nabla_x ||\nabla_{z^{(l)}} f||^2 + \frac{1}{4}\nabla_x ||\nabla_{z^{(l)}} f||^2 \otimes \nabla_x ||h^{(l-1)}||^2$$

The contribution from the bias is comparatively simple:

$$H_b = J_z J_z^\top \tag{75}$$

A.4 Derivation of the Correlation Lengthscale from the Cauchy Approximation

To determine the level sets of the Cauchy Approximation, we must solve:

$$C_{NTK}(x, x+u) = \frac{2a_x^2 + u^\top D_x}{2a_x^2 + u_x^\top D + u^\top H_x u} = c \tag{76}$$

Rearranging, and collecting terms, we have:

$$2a_x^2 + u^\top D_x - c(2a_x^2 + u_x^\top D + u^\top H_x u) = 0 \tag{77}$$

$$\Rightarrow 2(1-c)a_x^2 - c\Big(-\frac{1-c}{c}u^\top D_x + u^\top H_x u\Big) = 0 \tag{78}$$

$$\Rightarrow u^\top H_x u - \frac{1-c}{c}u^\top D_x = \frac{2(1-c)}{c}a_x^2 \tag{79}$$

$$\Rightarrow \Big(u - \frac{1-c}{2c}H^{-1}D\Big)^\top H\Big(u - \frac{1-c}{2c}H^{-1}D\Big) - \frac{(1-c)^2}{4c^2}D^\top H^{-1} D = \frac{2(1-c)}{c}a_x^2 \tag{80}$$

$$\Rightarrow \frac{\Big(u - \frac{1-c}{2c}H^{-1}D\Big)^\top H\Big(u - \frac{1-c}{2c}H^{-1}D\Big)}{\frac{2(1-c)}{c}a_x^2 + \frac{(1-c)^2}{4c^2}D^\top H^{-1}D} = 1 \tag{81}$$

This is the equation of an ellipse centred at $u = \frac{1-c}{2c}H^{-1}D$, and with shape matrix:

$$\Sigma_{shape} = \frac{H}{\frac{2(1-c)}{c}a_x^2 + \frac{(1-c)^2}{4c^2}D^\top H^{-1}D} \tag{82}$$

The area of this ellipse is (noting that H is a 2x2 matrix):

$$A_{ellipse} = \frac{\pi}{\sqrt{\det \Sigma_{shape}}} \tag{83}$$

$$= \frac{1}{\sqrt{\det H}}\Big(\frac{2(1-c)}{c}a_x^2 + \frac{(1-c)^2}{4c^2}D^\top H^{-1}D\Big) \tag{84}$$

The correlation lengthscale is then obtained from:

$$\xi = \sqrt{A_{ellipse}/\pi} \tag{85}$$

A.5 Minimum Value of C_{NTK}

We consider minimizing the following function:

$$f(u) = \frac{Q(u)}{P(u)} \tag{86}$$

$$Q(u) = 2a^2 + u^\top D \tag{87}$$

$$P(u) = Q(u) + u^\top Hu \tag{88}$$

Here, H is non-degenerate and positive definite. Thus:

$$\frac{\partial f}{\partial u} = \frac{\partial_u QP - Q\partial_u P}{P^2} = 0 \tag{89}$$

$$\implies \partial_u QP = Q\partial_u P \tag{90}$$

Thus:

$$(u^\top Hu)D = (4a^2 + 2u^\top D)Hu \tag{91}$$

Clearly $u = 0$ is a solution, and knowing that our expression locally approximates the cosine, we expect this to be a maximum. To find the other solution, which will be a minima, we take the dot product of both sides of the above equaiton with u. After simplifying, we obtain:

$$u^\top D = -4a^2 \tag{92}$$

If we insert this into equation 91, we get:

$$(u^\top Hu)D = -4a^2 Hu \tag{93}$$

$$\Rightarrow (u^\top Hu)H^{-1}D = -4a^2 u \tag{94}$$

$$\Rightarrow (u^\top Hu)(D^\top H^{-1}D) = 16a^4 \tag{95}$$

$$\Rightarrow u^\top Hu = \frac{16a^4}{DH^{-1}D} \tag{96}$$

Armed with an expression for $u^\top D$ and $u^\top Hu$, we derive the following formula for the min:

$$f_{min} = \left.\frac{2a^2 + u^\top D}{2a^2 + u^\top D + u^\top Hu}\right|_{u=\text{argmin} f} \tag{97}$$

$$= \frac{2a^2 - 4a^2}{2a^2 - 4a^2 + \frac{16a^4}{DH^{-1}D}} \tag{98}$$

$$= \frac{DH^{-1}D}{DH^{-1}D - 8a^2} \tag{99}$$

A.6 Relating Loss Gradient Variance to the NTK

Our goal is to quantify the amount of noise in the gradients of the local loss $\mathcal{L}(x_i) = \frac{1}{2}r(x_i;\theta)^2$. We have, in terms of the Jacobian $J_{ip} = \partial_{\theta_p} f(x_i)$, the following sample matrix for the gradients:

$$G = RJ \tag{100}$$

Here we have defined:

$$R = \text{diag}(r) \tag{101}$$

For a dataset with N samples, the sample mean and covariance are given by:

$$\mu = \frac{1}{N}G^\top 1_N \tag{102}$$

$$= \frac{1}{N}J^\top r \tag{103}$$

$$C = \frac{1}{N}J^\top R^2 J - \mu\mu^\top \tag{104}$$

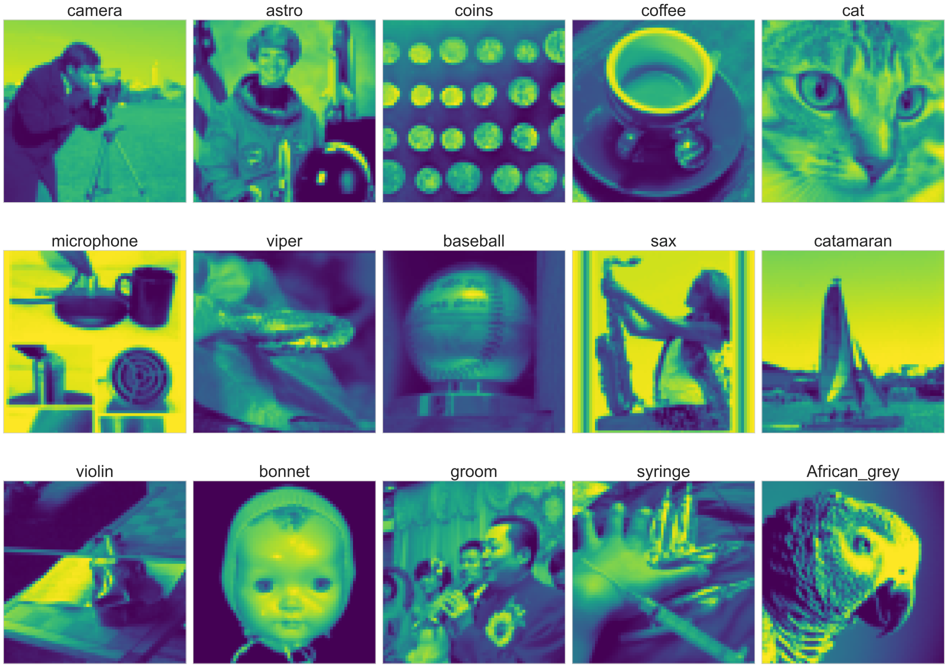


Figure 9: **Images used for training INRs**

From the cycle property of the trace, we have:

$$\text{tr}(J^\top R^2 J) = \text{tr}(R^2 J J^\top) \tag{105}$$

$$= \text{tr}(R^2 K_{NTK}). \tag{106}$$

We also have:

$$\text{Tr}(\mu\mu^\top) = ||\mu||^2 \tag{107}$$

$$= \frac{1}{N^2} r^\top J J^\top r \tag{108}$$

$$= \frac{1}{N^2} r^\top K_{NTK} r \tag{109}$$

Thus the variance of the loss gradients is given by:

$$\sigma_\theta^2 = \frac{1}{N}\text{Tr}(R^2 K_{NTK}) - \frac{1}{N^2} r^\top K_{NTK} r \tag{110}$$

B Experimental Details

B.1 Model Training

All our SIREN models are trained on the images shown in Figure 9, which we obtain through the python package scikit-image (Van der Walt et al., 2014), and the ImageNet dataset (Deng et al., 2009). These images are down-sampled to a resolution of 64×64 for training, but as a validation task, we track the reconstruction error on the images downsampled to 256×256 resolution. Our SIREN models are implemented using Pytorch (Paszke et al., 2019), and trained using NVIDIA RTX A6000 48GB GPUs for 10000 epochs, using full batch gradient descent with a learning rate of 1e-3. In our experimental sweeps, we consider the following ranges:

- Random seeds from interval $[0, 5]$.
- Width from set $\{64, 128\}$.
- Depth from set $\{3, 4, 5\}$.
- ω_0 from set $\{15, 30, 60, 90\}$.

B.2 Order Parameter Estimation

Analytical Order Parameters: To compute the NTK, we use a manual implementation of backpropagation to compute the gradients $\nabla_{z^{(l)}} f(x)$ for each layer, along with the hidden activations $h_{(l)}(x)$. The NTK is then constructed efficiently using the decomposition across layers outlined in Section A.1. To evaluate the local structure components a and D defined in equations 16 and 17, we obtain the spatial gradients using functorch (Horace He, 2021). We also assemble the H defined in equation 18 in this way, except we leverage the decomposition outlined in Section A.3.2 to streamline this process, and occupy less memory.

Empirical Order Parameters: Below we describe the estimation procedure for each of the empirical order parameters.

- To estimate the correlation functions empirically, we group pairs of datapoints into 50 bins based on a uniform division of the range of distances. Based on the coordinate range, the minimum distance is 0, and the maximum distance is $2\sqrt{2}$. Within each bin, we evaluate the mean of the C_{NTK}, defining $c(\epsilon)$. Based on these groups, we estimate our order parameters as follows:
  - To estimate the asymptotic value c_∞, we compute the mean value of $c(\epsilon)$ over the last ten bins (corresponding to points with the furthest separation).
  - Given the asymptotic value, we rescale all $c(\epsilon) \to \tilde{c}(\epsilon) = \frac{c(\epsilon)}{1-c_\infty}$, and then use linear interpolation to find the value of ϵ for which $\tilde{c}(\epsilon) = 0.5$, the FWHM. We then have $\xi_{corr} = \frac{\text{FWHM}}{\sqrt{2 \ln 2}}$.
- As an additional measure of the correlation length-scale (which we will use in Appendix D), we may calculate the number of points N_C for which C_{NTK} is greater than some cutoff (we use $\frac{1}{2}(1 + c_\infty)$). The effective correlation lenght-scale is then given by $\sqrt{N_C dA/\pi}$, where dA is the area of the coordinate grid cells. We denote this estimate ξ_{FWHM}.
- To estimate AUC$(v_0, \nabla I)$, the ground truth edges are identified using the Canny Edge Detector distributed through scikit-image (Van der Walt et al., 2014). We then evaluate the Area Under the Receiver Operating Characteristic Curve (ROC AUC) using the implementation in scikit-learn (Pedregosa et al., 2011). The principal eigenvector v_0, and the principal eigenvalue λ_0, are both computed using our own implementation of the Randomized Singular Value Decomposition built with pytorch (Paszke et al., 2019), using 3 iterations and 10 oversamples.
- To evaluate the Centred Kernel Alignment, in order to prevent zero modes from obscuring alignment, the following centred-variant of the normalized Hilbert-Schmidt Information Criterion (HSIC) is employed:

$$\text{CKA}(K, K') = \frac{\text{Tr}(K_c K_c')}{\sqrt{\text{Tr}(K_c K_c)\text{Tr}(K_c' K_c')}} \tag{111}$$

  Here, K_c denotes that a centrering operation has been applied, and is defined as:

$$K_c = (I - \frac{1}{n}11^\top)K(I - \frac{1}{n}11^\top) \tag{112}$$

  For both K_X and K_Y, we use bandwidths $\kappa = 0.1$.

Identifying Critical Points:

- For the gradient variance σ_θ^2 and the loss rate $\dot{L}_{\text{eval}}$, the location, and confidence region, for the critical points are identified using the peak detection algorithm distributed through scipy.signal (Virtanen et al., 2020). We filter for peaks with prominence greater than 0.5.
- For the $\min C_{NTK}$, we linearly interpolate to find the time t where $\min C_{NTK}$ crosses 0. To compute the confidence interval, we also track the cumulative std of $\min C_{NTK}$, denoted $\epsilon(t)$. We then use the same linear interpolation strategy to find the times where $\min C_{NTK} = \epsilon(t)$ and $\min C_{NTK} = -\epsilon(t)$.

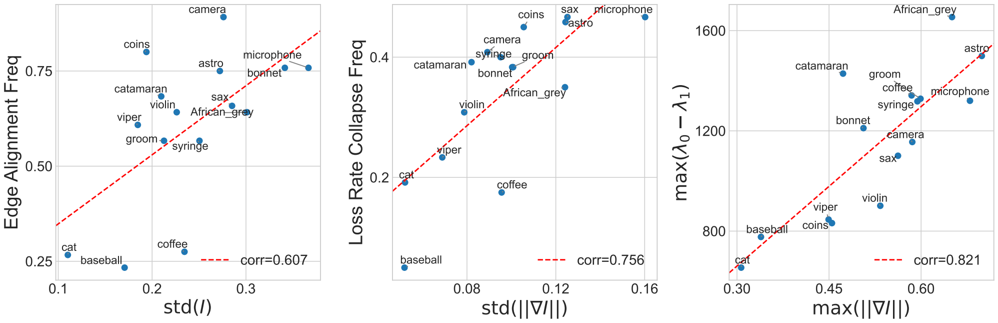


Figure 10: **Image Properties Affect Detection of Phase Transitions**. Left: Impact of the image variance on NTK alignment, as measured by occurrence of edge alignment. Center: Impact of variance in the image gradient on the occurrence of loss rate collapse. Right: Impact of the maximum image gradient on the spectral gap, another measure of NTK alignment. Line of best fit is shown as a red dashed line, with Pearson correlation displayed in the legend.

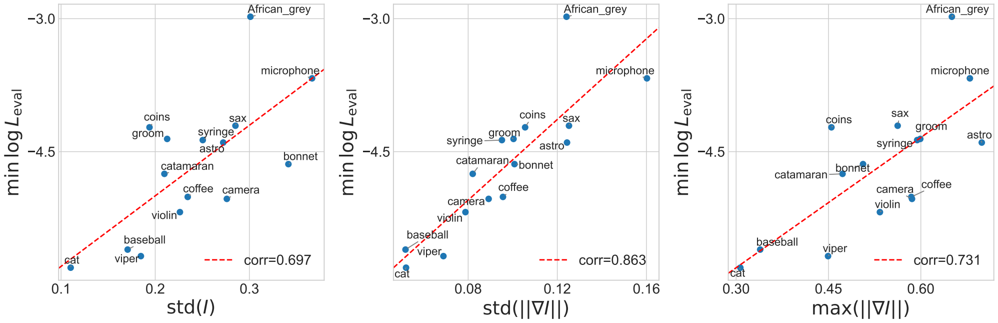


Figure 11: **Image Properties Affect Model Performance**. Plotting measures of image complexity against the best super-resolution reconstruction score. Line of best fit is shown as a red dashed line, with Pearson correlation displayed in the legend. We see a strong correlation between the std of the magnitude of the image gradients, and the performance achievable with our hyperparameter sweep.

- For all other parameters, we fit a sigmoid using the curve fitting function from scipy.optimize, with the default settings. The curve we fit has the form:

$$f(x; A, B, \mu, w) = A + (B - A)\left(1 + e^{-(x-\mu)/w}\right)^{-1} \tag{113}$$

  We identify the time $t = \mu$ with the critical point, with confidence region defined by $\mu \pm 2w$. For MAG-Ma, the critical point occurs when the order parameter begins to deviate from 0. Thus, we take $\mu - 2w$ as the critical point, with confidence interval $[\mu - 3w, \mu - w]$.

C Occurrence Rates of Phase Transitions

C.1 Impact of Image Features

There are three main cases in which a critical point cannot be reliably identified in an order parameter trajectory:

1. Peaks in the gradient variance σ_θ may be absent, or not prominent enough, to be detected using a standard peak detector. This happened in 21% of our experiments.
2. A zero-crossing cannot be found for the $\min C_{NTK}$ because, at initialization, it is already less than 0. This happened in 51% of our experiments.

3. Edges do not feature prominently in the principal eigenvector v_0 of the NTK, so that AUC$(v_0, \nabla I)$ remains close to 0.5 throughout training. This happened in 39% of our experiments.

It is important to note, phase transitions may still occur even during these failure modes - the shift in the order parameter may be simply too weak to be identified by the change detection algorithm outlined in Section . It is for this reason that we employ multiple order parameters to identify the same transition, For example, AUC$(v_0, \nabla I)$, λ_0, and CKA(K_Y, K_{NTK}) all exhibit critical behaviour during NTK Alignment[4].

Nevertheless, it is instructive to identify what properties of image datasets may be used to predict the aforementioned failure modes. To this end, for each experimental run, we determine if any of the previously mentioned failure modes has occurred, and then record the frequency of success for each image studied. In Figure 10, we see that these frequencies correlate with the complexity of the image, as measured by its variance, and the variance of its gradient magnitude. These same properties correlate strongly with the best model performance achieved across all hyperparams (Figure 11). These correlations give additional support to the mechanism described in Section 3.3, whereby SIREN models struggle to fit edges as they have sharp gradients. In this way, NTK alignment may be a symptom that the model is struggling to learn from the data, and has begun the process of memorization. As other authors have postulated, memorization occurs at the transition between the fast and slow learning phases (Shwartz-Ziv & Tishby, 2017), which accounts for the coincidence between these two transitions.

Finally, on the right side of Figure 10, we investigate a potential mechanism by which image properties induce these transitions. Our analysis in Section 3.3 suggests that an important prerequisite for edge alignment to occur is the divergence of the principal eigenvalue in comparison to the next leading eigenvalue (the spectral gap). We see that the max value of the spectral gap across all experimental runs correlates strongly with the maximum magnitude of the image gradient.

C.2 Impact of Hyperparameters

The broad effects of varying depth and ω_0 on AUC$(v_0, \nabla I)$ are summarized in Table 1. To gain deeper insight into how these parameters influence the principal eigenvector, we examine the two case studies illustrated in Figure 12. A larger ω_0, by narrowing the correlation lengthscale, increases the number of points the model memorizes. Initially, this results in an increase to AUC$(v_0, \nabla I)$, but as ω_0 grows, noise is memorized as well, obscuring the edges. By contrast, increasing depth makes v_0 less uniform, concentrating it around the edges.

[4] Also note, even when AUC$(v_0, \nabla I)$ is weak, edges are still visible in the principal eigenvector, as seen in Figure 12

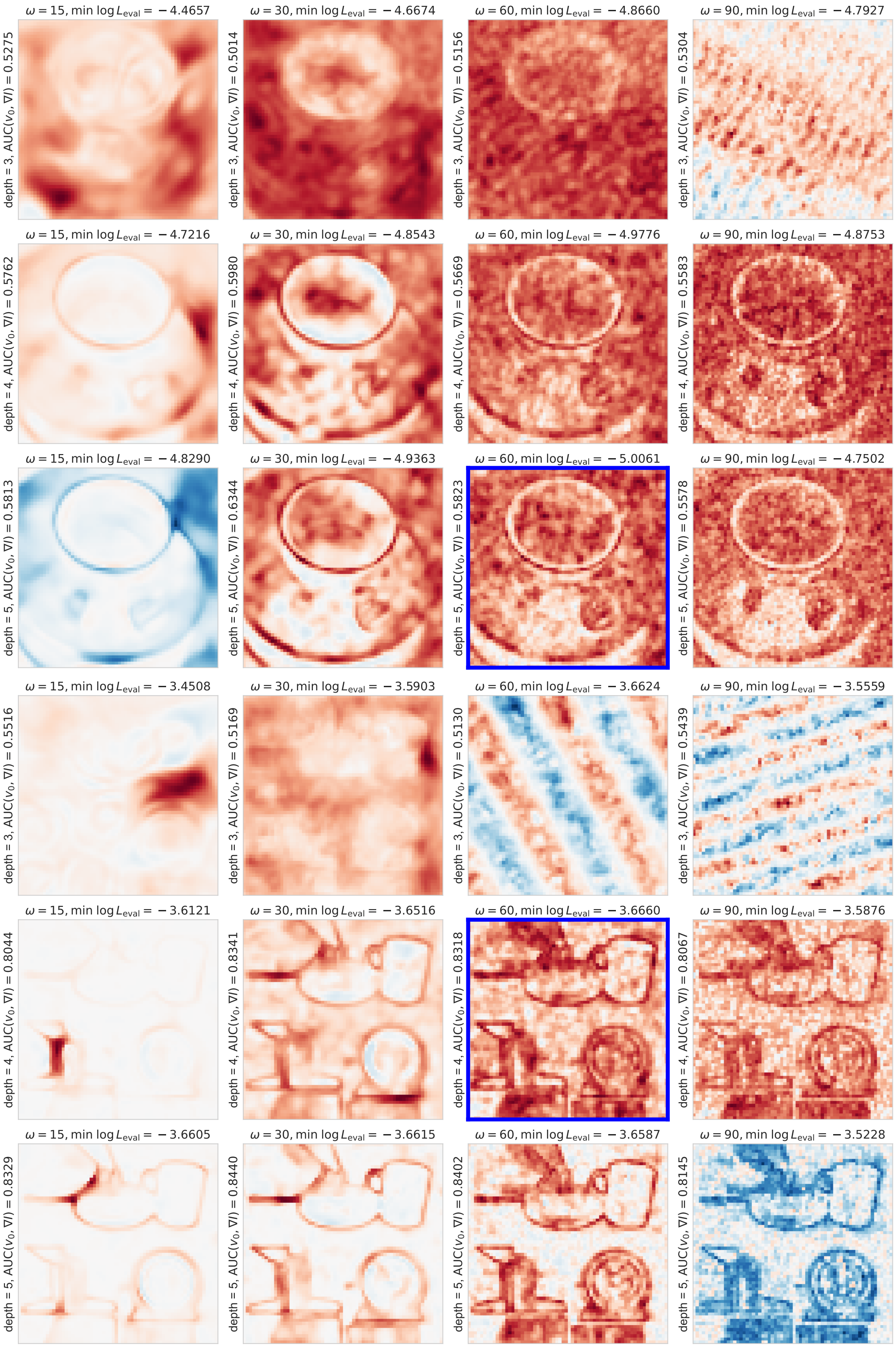


Figure 12: **Variation in NTK Alinment with Hyperparameters**. Principle eigenvectors of the NTK at the end of training. Top three rows: the coffee dataset. Bottom three rows: the sax dataset. Best performing architecture highlighted in blue.

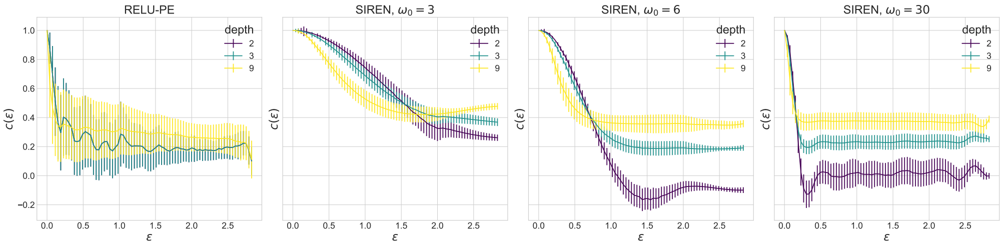


Figure 13: **Effect of Hyperparameters on Correlation Functions At Initialization**: In ReLU-PE models, the Gaussian approximation of the C_{NTK} correlation function is poor for all depths, due to high-variance, long range interactions. By contrast, for SIREN models, there is much less variance, and the range of the interactions shrinks for increasing ω_0.

D Comparison with ReLU Activations

To justify our focus on sinusoidal neural networks, in this section we examine the learning dynamics of ReLU-MLPs, based on the positional encoding scheme used in (Mildenhall et al., 2020). The positional encoding layer is kept static, and we pre-compute the nyquist frequencies corresponding to our image size (64×64), as is done in (Sitzmann et al., 2020). We denote this architecture ReLU-PE. All other architectural choices are identical to those described in Appendix B. We observe a number of differences between SIRENs and ReLU-PEs (visualized in Figure 14):

- Firstly, SIREN models exhibit strong locality: over the course of training, the asymptotic value of the C_{NTK} decays to 0, whereas it grows in ReLU-PE models. What's more, the range of interaction as measured by ξ_{FWHM} is larger in ReLU-PE models. An example comparing the correlation functions for both architectures is shown in Figure 13.
- Secondly, learning is much slower in ReLU-PE models than it is in SIRENs. One explanation for this is that there is more gradient confusion (Sankararaman et al., 2020), that is, the minimum value of the C_{NTK} is lower. In particular, $\min C_{NTK}$ is less than zero across all ReLU-PE runs, so that these models are always operating in the ”slow” phase of learning.
- The principal eigenvalue λ_0 of the NTK grows to be orders of magnitude larger for SIREN models than for ReLU-PE models. That said, tangent kernel alignment still occurs in ReLU-PE models, it is just a much slower process. In Figure15, we train a 7-layer deep, 128-unit wide MLP full-batch with a learning rate of $1e-3$ for 250k epochs, varying only the activation function. To reach the edge-alignment achieved by a SIREN model after 453 epochs, the ReLU-PE model must train for 239986 epochs. We also see that more of the edges are present in the principal eigenvector of the SIREN model's NTK.
- At initialization, MAG-Ma is orders of magnitude lower for SIREN models than for Relu-PE models, indicating the latter are already operating in a phase where translational symmetry is broken.

In summary, while ReLU-PE models exhibit Neural Tangent Kernel alignment, it is a much slower, non-local process, that does not coincide with loss-rate collapse or translational symmetry breaking.

E Implications of Local Image Structure on Feature Learning

We are now positioned to elucidate the features learned during NTK alignment. As proposed in the previous section, the local structure of the NTK adapts to the spatial variations in parameter gradients. In this section, we delve into the spectral consequences of this adaptation. We contend that the principal eigenvectors evolve into edge detectors, resembling the auto-correlation structure tensors commonly employed in traditional computer vision. This observation reinforces the concept of translation symmetry breaking: in computer vision, the utility of auto-correlation structure tensors stems from the premise that the most informative features are those that minimize redundancy. The auto-correlation function quantifies this through metrics of translational symmetry breaking.

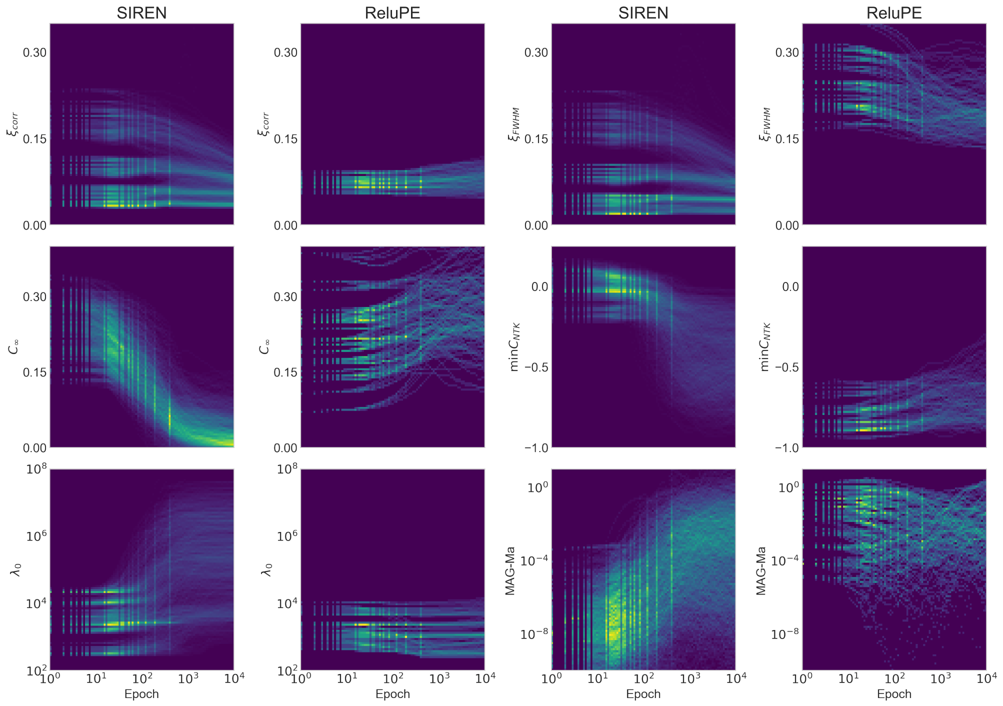


Figure 14: **Learning Trajectories for SIREN and ReLU-PE models**: Histograms visualizing the distribution of various order parameters throughout training. See Section B for full details on models and datasets used.

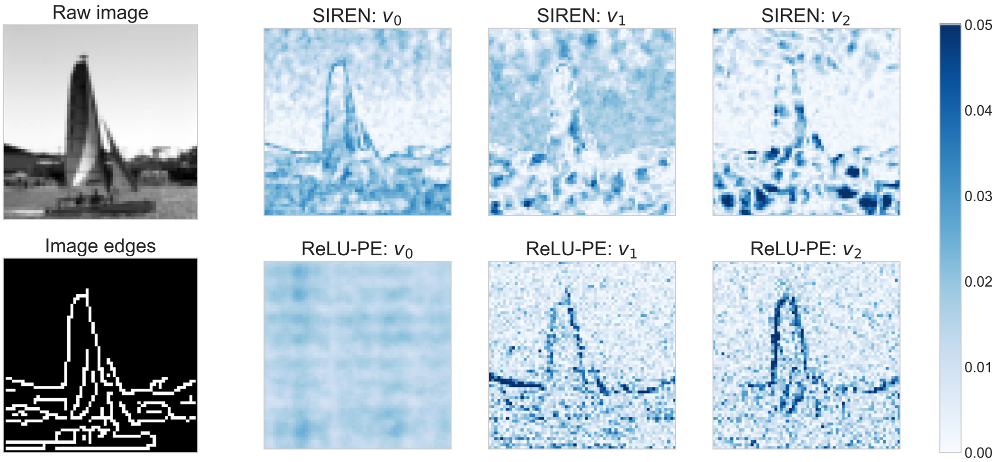


Figure 15: **NTK alignment in SIREN and ReLU-PE models**: The principal eigenvectors of the NTK at the end of training. Final AUC$(v_1, \nabla I)$ for the ReLU-PE is 0.754, whereas AUC$(v_0, \nabla I)$ for the SIREN model is 0.804. The training time required to achieve an edge-alignment score greater than 0.75 for the SIREN model was 453 epochs, whereas for the ReLU-PE model it was 239986 epochs.

Per the discussion in Section 3.3, the principal eigenvector is closely related to the auto-correlation function. By leveraging the decomposition of the NTK in equation 36, we may relate to the features considered in computer vision. Let us define:

$$w^{(l)}(u; x) = 1 + h^{(l-1)}(x)^\top h^{(l-1)}(x + u) \tag{114}$$

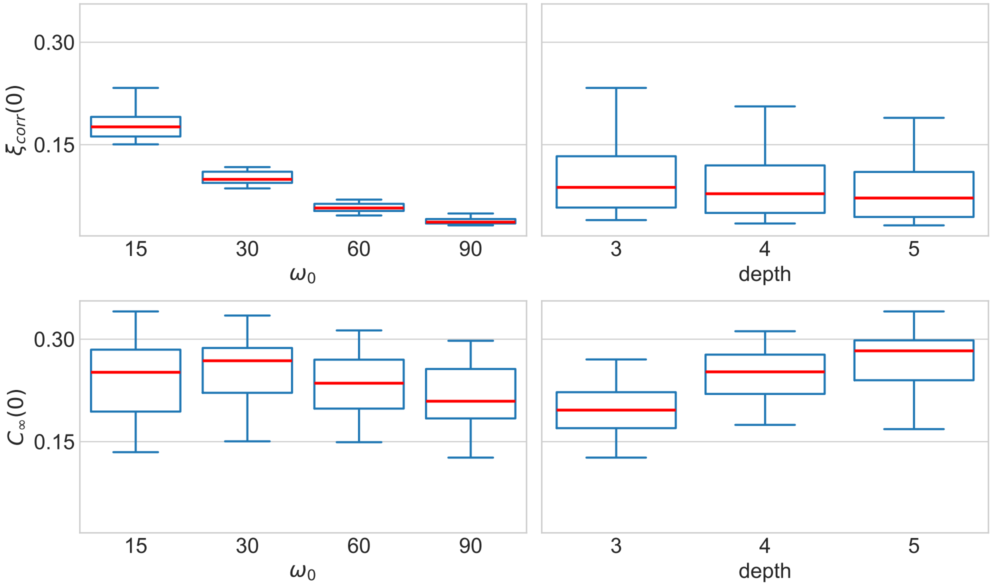


Figure 16: **Hyperparameters Affect Local NTK Structure**. Boxplots visualizing the distribution of structural parameters for the C_{NTK}. Top row: variation in the initial correlation lengthscale $\xi_{corr}(0)$. Bottom row: variation in the initial asymptotic value of the C_{NTK} ($C_\infty(0)$).

so that the largest contribution comes from the immediate neighbourhood of x. This motivates us to perform a Taylor expansion of the remaining terms as follows:

$$K_l 1 = \sum_u K_l(x, x+u) \tag{115}$$

$$= \sum_u w_l(u;x) \sum_d \frac{\partial f(x)}{\partial z_{ld}} \frac{\partial f(x+u)}{\partial z_{ld}} \tag{116}$$

$$\approx \sum_u w_l(u;x) \sum_d \left(\frac{\partial f(x)}{\partial z_{ld}} \frac{\partial f(x)}{\partial z_{ld}} + h.o.t \right) \tag{117}$$

$$= \text{tr}(A_l(x_i)) + h.o.t \tag{118}$$

Here, A_l denotes the structure tensor used in the Harris-Corner detector (Harris & Stephens, 1988). Accordingly, we see that $K1$ - and thus, the principle eigenvector - assess the extent of local translational symmetry disruption near a point x. This principle underlies feature selection in computer vision, a concept mirrored in NTK feature learning, as evidenced by the principal eigenvectors that are predominantly maximized around dataset edges and corners.

It is crucial to highlight that A_l pertains to the structure tensor of a specific layer l. Collectively, the entire DNN's NTK facilitates feature selection across a scale pyramid.

F Evaluating Fidelity of Approximation

F.1 Local Structure of the NTK

As described in Section 3.1, INRs are often carefully designed to ensure a diagonally dominant NTK (Tancik et al., 2020; de Avila Belbute-Peres & Kolter, 2022; Liu et al., 2023). In higher dimensions, diagonal dominance is equivalent to a bias towards local interactions. We see in Figure 16 the hyperparameters that most affect this local structure: we observe that, while depth has a small impact on the initial correlation lengthscale $\xi_{corr}(0)$, higher values of ω_0 cause the C_{NTK} to become

depth	ω_0	ξ	$\min C_{NTK}$	v_0
3	15	0.984 ± 0.010	0.897 ± 0.065	0.977 ± 0.012
	30	0.867 ± 0.155	0.915 ± 0.053	0.983 ± 0.007
	60	0.652 ± 0.444	0.941 ± 0.043	0.987 ± 0.004
	90	0.733 ± 0.346	0.966 ± 0.020	0.986 ± 0.006
4	15	0.968 ± 0.031	0.948 ± 0.037	0.979 ± 0.008
	30	0.734 ± 0.190	0.965 ± 0.034	0.984 ± 0.007
	60	0.684 ± 0.386	0.961 ± 0.033	0.984 ± 0.010
	90	0.941 ± 0.092	0.963 ± 0.033	0.983 ± 0.009
5	15	0.955 ± 0.044	0.939 ± 0.035	0.975 ± 0.010
	30	0.780 ± 0.146	0.963 ± 0.032	0.980 ± 0.008
	60	0.828 ± 0.237	0.969 ± 0.034	0.979 ± 0.038
	90	0.967 ± 0.031	0.976 ± 0.031	0.980 ± 0.011

Table 2: **Fidelity of Cauchy Approximation**: Pearson correlation between the true order parameter and predictions using the local Cauchy Approximation. Mean and standard deviation are calculated over the spread of models and datasets described in Section B.

dramatically more localized. The converse is true for the asymptotic value $C_\infty(0)$: ω_0 has a minor effect, but increasing depth leads to stronger interactions across large distances.

F.2 Cauchy Approximation

To ascertain the fidelity of the Cauchy Approximation, we estimate the Pearson correlation between the true values of the correlation lengthscale ξ and $\min C_{NTK}$, and the prediction based only on the local model. We choose this metric because the identification of critical points is insensitive to linear transformations. The results are shown in Table 2. Similarly, we evaluate our approximation of the principle eigenvector v_0, by looking at the absolute cosine distance between our approximation and the ground-truth.

Finally, in Section 3.3, we approximated the principal eigenvector v_0 of the NTK K with the row mean $K1/1^\top 1$. The median cosine alignment between the row mean and the true v_0 was found to be 0.99995 across all epochs surveyed, across all models and datasets. The IQR is 0.00446. The strength of this approximation is a testament to the extreme spectral gap of the NTK, which itself is a consequence of NTK alignment.

G Additional Experimental Results

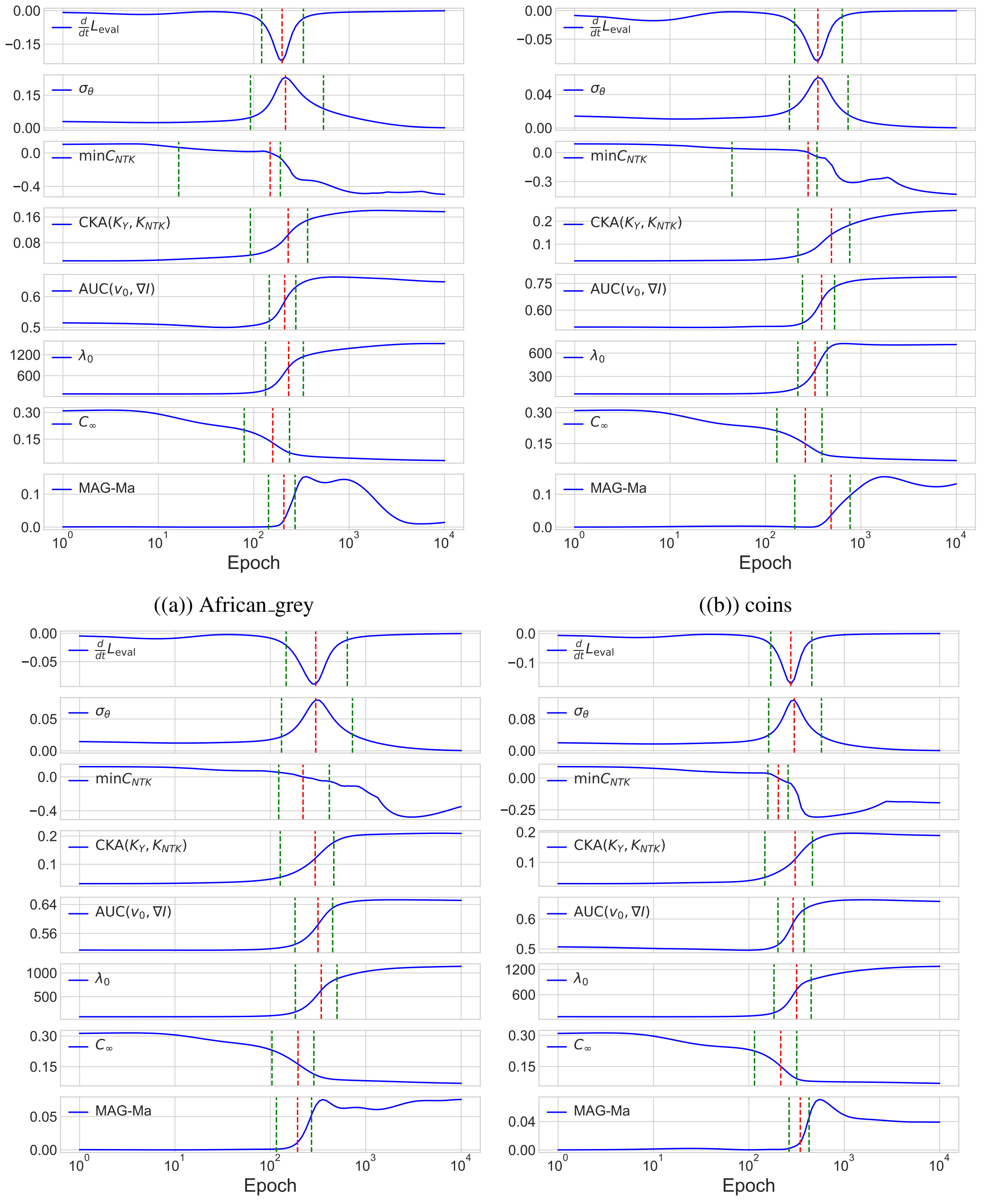
G.1 Order Parameter Trajectories for Single Runs

This section contains additional illustrations of the order parameter trajectories, and the corresponding confidence region estimates, similar to the left side of Figure 4. The results are shown in Figure 17. Each model is a 5 layer deep, 128-unit wide SIREN network, trained with full-batch gradient-descent with a learning rate of 1e-3.

G.2 Influence of Hyperparameters on Order Parameter Trajectories

In this section, we show that the analysis in Section 4.3 generalizes beyond the cameraman dataset. Figures 18-20 showcase the effect of depth. Figures 21-23 showcase the effect of the bandwidth parameter ω. The experimental settings are identical to those described in Section 4.3, just applied to different datasets.

We additionally track the CKA between the NTK and a static RBF kernel with fixed bandwidth K_X, as described in 4.1. The evolution of this hyperparameter reflects the evolution of the correlation lengthscale ξ_{corr}. When this value is large (as it is when ω_0 is small), the NTK has a broad diagonal, and thus overlaps well with the RBF. Over the course of training, ξ_{corr} shrinks, and thus, so does CKA(K_X, K_{NTK}).



((a)) African_grey

((b)) coins

((c)) groom

((d)) syringe

Figure 17: **Alignment of Order Parameters**. Order parameter evolution and critical points during training of a SIREN model. The red vertical lines denote the location of the critical points, and the green vertical lines denote confidence regions.

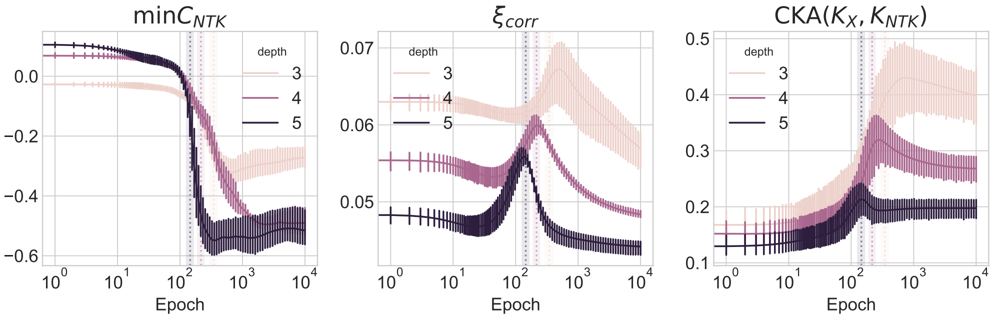


Figure 18: **Effect of depth on Critical Behaviour (Microphone)**: Average MSEs, in order of ascending depth: $2.561e^{-2} \pm 9.355e^{-5}$, $2.555e^{-2} \pm 8.970e^{-5}$, $2.572e^{-2} \pm 7.209e^{-5}$. Dashed vertical lines denote the location of the peak of the loss rate $\dot{L}_{\text{eval}}$, marking the phase transition.

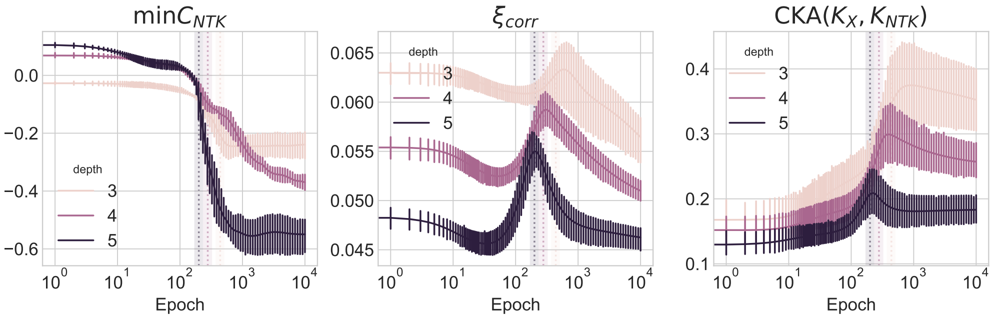


Figure 19: **Effect of depth on Critical Behaviour (Sax)**: Average MSEs, in order of ascending depth: $1.628e^{-2} \pm 1.312e^{-4}$, $1.513e^{-2} \pm 3.384e^{-5}$, $1.494e^{-2} \pm 6.605e^{-5}$. Dashed vertical lines denote the location of the peak of the loss rate $\dot{L}_{\text{eval}}$, marking the phase transition.

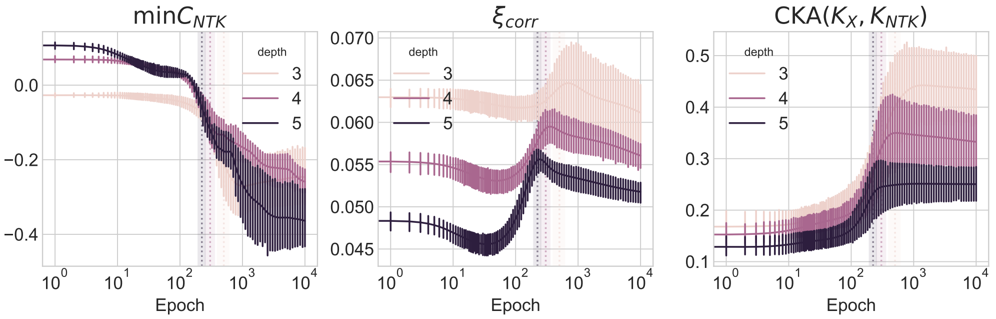


Figure 20: **Effect of depth on Critical Behaviour (Violin)**: Average MSEs, in order of ascending depth: $6.885e^{-3} \pm 1.677e^{-4}$, $5.930e^{-3} \pm 5.016e^{-5}$, $5.665e^{-3} \pm 3.640e^{-5}$. Dashed vertical lines denote the location of the peak of the loss rate $\dot{L}_{\text{eval}}$, marking the phase transition.

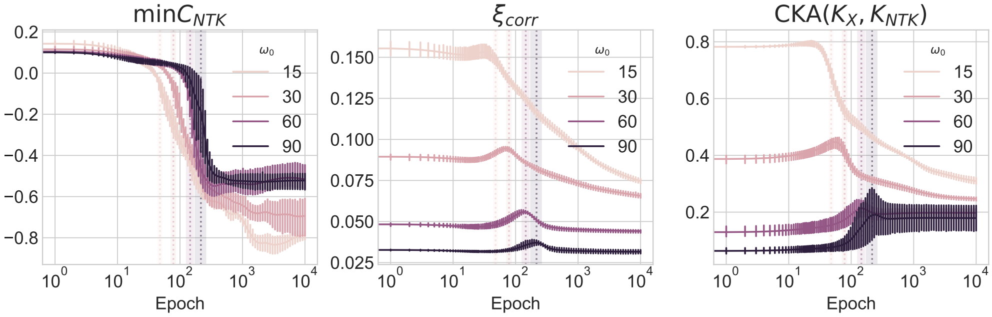


Figure 21: **Effect of ω_0 on Critical Behaviour (Microphone)**: Average MSEs, in order of ascending ω_0: $2.601e^{-2} \pm 1.804e^{-4}$, $2.566e^{-2} \pm 1.327e^{-4}$, $2.572e^{-2} \pm 7.209e^{-5}$, $2.807e^{-2} \pm 7.688e^{-4}$. Dashed vertical lines denote the location of the peak of the loss rate $\dot{L}_{\text{eval}}$, marking the phase transition.

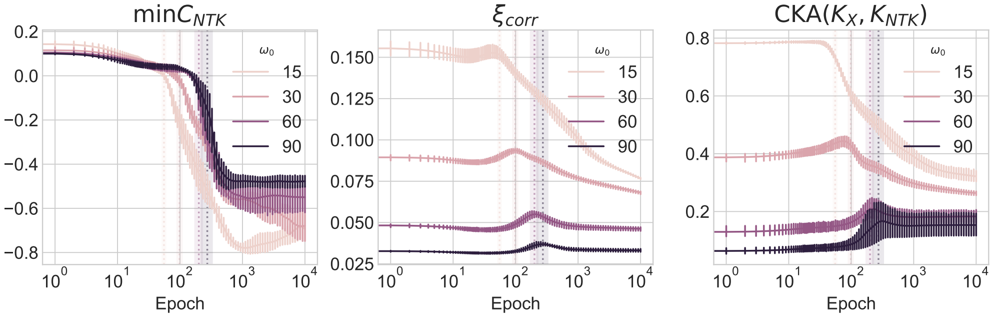


Figure 22: **Effect of ω_0 on Critical Behaviour (Sax)**: Average MSEs, in order of ascending ω_0: $1.680e^{-2} \pm 1.666e^{-4}$, $1.561e^{-2} \pm 6.552e^{-5}$, $1.494e^{-2} \pm 6.605e^{-5}$, $1.639e^{-2} \pm 3.938e^{-4}$. Dashed vertical lines denote the location of the peak of the loss rate $\dot{L}_{\text{eval}}$, marking the phase transition.

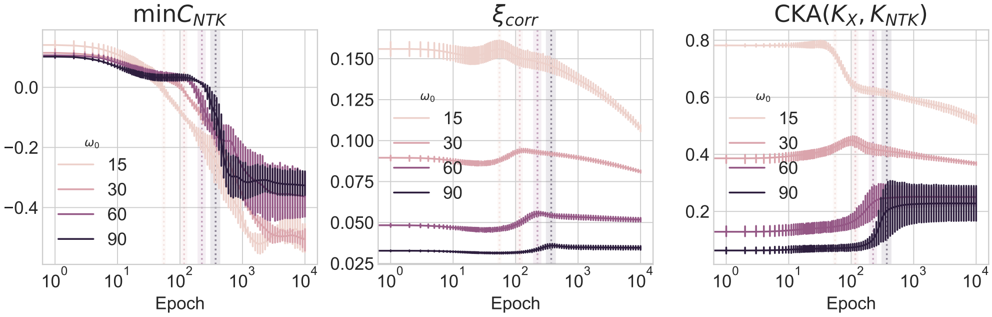


Figure 23: **Effect of ω_0 on Critical Behaviour (Violin)**: Average MSEs, in order of ascending ω_0: $7.223e^{-3} \pm 1.503e^{-4}$, $6.305e^{-3} \pm 3.139e^{-5}$, $5.665e^{-3} \pm 3.640e^{-5}$, $6.698e^{-3} \pm 3.359e^{-4}$. Dashed vertical lines denote the location of the peak of the loss rate $\dot{L}_{\text{eval}}$, marking the phase transition.